CHAPTER

12

SECURING NETWORK RESOURCES

After reading this chapter and completing the exercises, you will be able to:

- Explain how an operating system and applications are licensed
- Explain how to edit the rights and permissions associated with groups and create your own groups with special rights
- Explain the various methods of user authentication available in Windows 2000
- Distinguish between policies and profiles and explain how both are used in Windows 2000
- Discuss the encryption methods employed by Windows 2000

The point of a network operating system is to share resources with the other people using the network. Sometimes, however, you don't want everyone using the network to have access to every available resource. Perhaps they haven't paid for access to the resource, or perhaps you'd rather not allow a certain person or group to use a particular resource.

This chapter outlines the ways in which you can use an operating system to control access to network resources. We explain how an operating system and applications are licensed, and how to edit the rights and permissions associated with groups, or create your own groups with special rights. We also distinguish policies and profiles and explain how each are used in Windows 2000. Then, we explain the various methods of user authentication available in Windows 2000 and discuss the encryption methods employed by Windows 2000.

APPLICATION AND CLIENT LICENSING

When you purchase an operating system or an application, you don't so much purchase the software as the right to use that software.This is an important distinction. Buy a server operating system, for example, and you might get 10 **client access licenses (CALs)** with it, which means that up to 10 people are allowed to connect to the server running that operating system. If more than 10 people want to connect to the server, you need to buy more CALs. Applications work the same way. For example, when you buy a copy of any application, you purchase a license to use the application under certain conditions outlined within the software. You don't have the right to make that software available to additional users or to install it on as many computers as you like.The exact conditions of the license—the number of people who can use the software and under what conditions—depend on the wording of the application's **End User License Agreement (EULA)**.

Per-Seat Versus Per-User Licensing

Two basic categories (or modes) of licenses are available: per-session (also called per-user) and per-seat.

Per-session licenses allow a certain number of simultaneous connections (sessions) to the licensed operating system or application.That is, if your operating system comes with 10 per-user licenses, 10 people can log onto the server at once. If an eleventh person wants a connection, either he or she must wait for someone to log off the server or you must buy another user license.

Per-seat licenses work a little differently.These types of licenses apply to computers, rather than users.That is, a certain computer is given the right to access the licensed operating system or application. As many people as you like may use that computer for access, but only the computers with the licenses may be used.Therefore, if computers Alpha, Beta, Gamma, and Delta all have per-seat licenses to WordCruncher running on an application server, Jane can sit at any one of those computers to run WordCruncher. She cannot run WordCruncher from computer Epsilon, however, because that computer is not licensed.When access to an application or operating system is governed by a per-seat license, any computer that connects to that application or operating system needs a valid access license—even if it uses the application once only once per year.

Note

Specific types of licenses are available that can be either per-seat or per-session, as you will see in the next section, "Types of Licenses Required."

The two main classes of licenses each have their own advantages, but you can't choose the class of license that best fits your needs on the basis of its advantages or disadvantages. Licenses are sold on an "as-is" basis; that is, an application or operating system is licensed on a per-seat or per-session basis by its creator.You can't pick which kind of license you'd like to have.

Note

Occasionally, you may encounter an application or operating system that gives you an option of licensing by seat or by session. After you make the choice, you must stick with it, and you can't buy some licenses of one class and the rest of the other.

Types of Licenses Required

To access a Windows 2000 operating system, you might need any or all of four types of licenses: a console license, a client access license, a terminal server license, or an application license. All of these types of licenses may be either per-session or per-seat licenses, depending on the software involved.

Console License

The **console license** comes with an operating system and represents permission to install the operating system on a single computer and log onto it. Console licenses are by their nature per-seat; they allow you to install an operating system once and work with it at that computer. They do not allow you to install the software on as many computers as you like so long as you're the only person using the operating system.

Client Access Licenses

A **client access license** allows the bearer to access a server—in this case, a Windows 2000 Server. That access includes some use of the server's functionality; for example, Domain Name System (DNS) name resolution, the lease of an Internet Protocol (IP) address from a Dynamic Host Configuration Protocol (DHCP) server, or the ability to reach shared devices and resources on the network. Basically, a client access license permits you to use the core functionality of the Windows 2000 Server. If you plan to log onto a Windows 2000 Server from the network, you need a valid client access license.

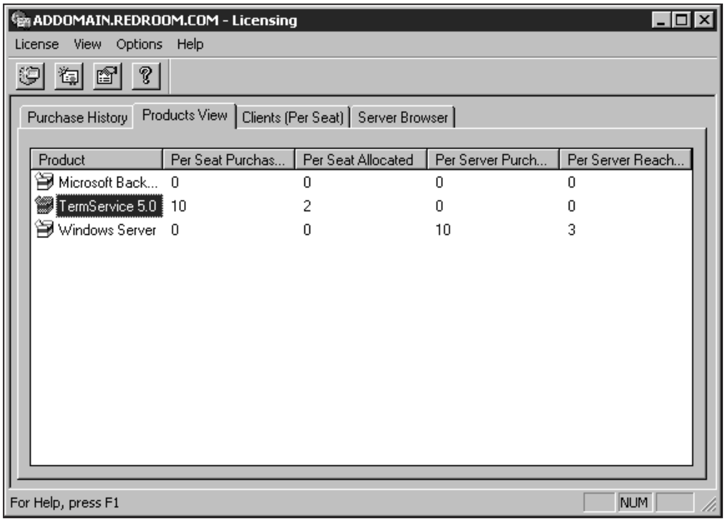
Windows 2000 includes a licensing tool that you can use to monitor the licenses available to the Windows 2000 and Windows NT Servers in the local or trusted **domains** on the network. Using this tool (shown in Figure 12-1), you can see the number of licenses available on each server in the network and the number that are currently in use. You can also keep records of purchasing licenses.

Terminal Server Licenses

Anyone accessing terminal server sessions (described in Chapter 11) needs one of two types of **terminal server client access license (TSCAL)**. People who are members of the Windows 2000 domain need a standard TSCAL to run a terminal services session, as well as a client access license to access the server in the first place. TSCALs are given on a per-seat basis, meaning that before people start logging onto the terminal server, you need to determine who has that right and who doesn't.

Note

Windows 2000 Professional comes with a TSCAL, so you can log onto a terminal server from this operating system without having to use one of the Windows 2000 server terminal server licenses. Other operating systems need an extra TSCAL.



Product	Per Seat Purchas...	Per Seat Allocated	Per Server Purch...	Per Server Reach...
Microsoft Back...	0	0	0	0
TermService 5.0	10	2	0	0
Windows Server	0	0	10	3

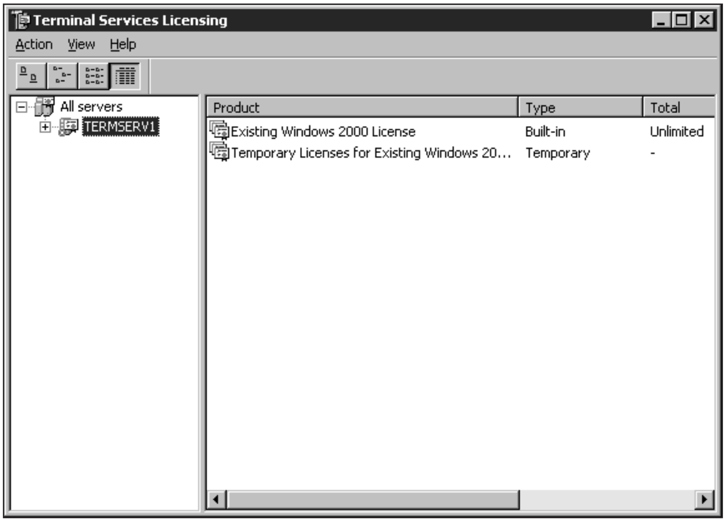
Figure 12-1 The Windows 2000 Licensing tool

People accessing the terminal server from the Internet who are not members of the Windows 2000 domain can pay for access to the terminal server with a per-session **Internet client license (ICL)**. Only anonymous users are allowed to log onto the terminal server via the ICL—not people with domain accounts. ICLs are intended for people accessing **application service providers (ASPs)**, companies that license applications via the Web for a fee. ICLs are really useless for any other purpose. For example, they cannot be used by companies to allow employees to access the terminal server from their homes via the Web.

To keep track of how TSCAL and ICLs are used, Windows 2000 supports a Terminal Services Licensing service and a related tool that you can use to monitor the license usage on each server running a licensing service (other than the regular Windows 2000 licensing). This tool is available from the Administrative Tools section of the Programs menu, as shown in Figure 12-2. The server running this service does not have to be running the Terminal Services software.

Application Licenses

Client access licenses and TSCALs apply only to operating system use. To use an application from that operating system, you need an **application license** that allows you to run the software. The way that the license works depends on the type of application involved. Single-user applications, such as word processors or spreadsheets, come with single licenses; applications that are used by several people at once, such as e-mail servers, come with a group license. These multiuser applications are sometimes called **groupware**.



Product	Type	Total
Existing Windows 2000 License	Built-in	Unlimited
Temporary Licenses for Existing Windows 20...	Temporary	-

Figure 12-2 The Terminal Services Licensing tool

USER AND GROUP RIGHTS AND PERMISSIONS

After you have all of the licenses required to access the resources on the network, obtaining the actual access still depends on the rights and permissions you have on the network. Those rights and permissions are controlled by the security management part of the operating system. In Windows 2000, they are governed by the Security Manager in the Windows 2000 executive.

Server Accounts Versus Domain Accounts

Everyone using network resources needs an account to log onto the network. This account can be either a server account or a domain account. Server accounts allow you to log onto a specific server, but do not grant access to any other network resources. The user account is located in a user account database stored on the server.

Domain accounts grant you access (or, more precisely, potential access) to the resources shared by all servers in the domain, which is the group of computers that rely on a centralized account database to secure resources. This centralized account database, which is stored on a server called a **domain controller**, is responsible for checking the credentials of anyone who attempts to log onto the domain. When the domain controller confirms that a user has an account on the domain, the user can browse the network resources on the domain. The user may or may not be able to use all resources on the computer; the user's access depends on the rights and permissions associated with his or her user account.

A domain can have more than one domain controller, which reduces the stress on the domain controllers when many people attempt to log onto the domain simultaneously. The presence of multiple domain controllers also prevents the domain from becoming inaccessible if one domain controller crashes and is unable to authenticate anyone. If you're running a mixed domain of Windows 2000 and other computers, the domain controller is responsible for authenticating your access to any network resources.

Note

A server account does not necessarily give you access to all shared resources on that server, nor does a domain account automatically ensure access to all shared resources on the domain. The kind of access that a user actually obtains depends on the rights and permissions attached to the user account and whether those rights are sufficient to access the objects in the domain.

USER AUTHENTICATION METHODS

Access to resources in Windows 2000 is based on **authentication**, which is the process of determining that you are who you say you are. When you type your user name and password to log onto a particular domain, WinLogon intercepts the information and passes it to the **Local Security Authority (LSA)**. If you're logging onto a single server with its own security database, the LSA checks whether you have an account on the local computer. If you're logging onto either the local domain or a trusted domain, the LSA communicates with that domain's domain controller to see whether you have an account in the domain controller's security database. If the LSA is able to confirm your identity (that is, if you have a valid user account), you are assigned a set of rights and permissions on the server or domain. If you don't have a valid account on the server or domain, you are denied access and not permitted to log on. Windows 2000 demands that you log on before you can use even the local computer; the security system does not allow you to bypass the LSA and just use local resources. The act of sitting at the keyboard and typing your name and password is called an **interactive logon**.

The LSA's job is not necessarily finished when an interactive logon is complete. At that point, you are authenticated at the computer you have logged into and are permitted to browse the network. If you attempt to connect to a resource on the domain, however, the LSA intercepts the request again and passes your credentials to the remote server to show that you're allowed to connect to that server.

As you can see, a lot of chatter takes place on the network as the LSA on your computer authenticates you at the other servers you want to access. The protocol used for this chatter in a LAN environment containing at least one pre-Windows 2000 operating system (including Windows NT 4, Windows 95/98, and Windows 3.1) is **NT LAN Manager (NTLM)**. Otherwise, the network uses **Kerberos**, which is the native Windows 2000 authentication protocol. The only exception occurs when you're being authenticated to a single computer—in other words, when you're logging onto a server, rather than a domain. In that case, NTLM is used even if you're logging onto a Windows 2000 server from a Windows 2000 Professional workstation. All computers on the network must use the same authentication protocol for remote communications, so if the network contains a single pre-Windows 2000 computer, then all computers must use NTLM authentication.

NTLM

NTLM is the default authentication protocol used in Windows NT 4. Basically, it makes sure that the client requesting a resource or service has permission to use that resource or service by communicating with a domain controller. Every time a client tries to access a server, the LSA consults the domain controller to verify that the client is permitted access. If access is permitted, the LSA grants access; if it is not, the client is denied access to the resource. The domain controller plays a part in all such transactions.

This method has some limitations. First, servers know that the clients are who they say they are, but the clients don't have similar knowledge about the servers. A rogue computer or service could impersonate a server and intercept requests from the client. Second, the domain controller must authenticate every communication between client and server, which leads to a lot of network traffic and can put unnecessary strain on the domain controller. For these reasons, Kerberos is a more secure and potentially faster authentication protocol.

Kerberos

Kerberos is the native Windows 2000 user authentication protocol that is designed for networks assumed to be insecure. Before any communication can take place between a client and a server—that is, between the computer requesting the resource and the computer that's got it—the identity of both client and server must be authenticated. Once a connection is established, the server and client can continue to use it until the client disconnects. If the client disconnects from the domain and then reconnects, the client and server must re-authenticate themselves.

Kerberos relies on a model of shared secrets. The idea is that if only two people (or computers, in this case) know a secret, they can use that secret to prove their identities to one another. For example, if Computer A wants to communicate with Computer B, Computer A must show Computer B that it knows the shared secret. Computer B then knows that it really is Computer A initiating the communication.

One way to communicate the secret is to include the secret in the network transmission. This technique, however, admits the possibility of a third computer listening on the same network, intercepting the message, and learning the secret. Thus, there must be way to use the secret without ever actually saying it.

Kerberos solves this problem by means of a cryptographic key known to both computers. Computer A demonstrates that it knows this key by encrypting the communication with it. Computer B demonstrates that it knows the key by decrypting the data and sending a receipt back to Computer A. If Computer B can't decrypt the data with the right key, it knows that the message wasn't really from Computer A. If Computer A receives a receipt, it knows that the message successfully reached the real Computer B. Because only the data encrypted with the key—not the key itself—actually goes on the network, a third party cannot intercept the key and use it. This process is called **secret key communication**.

Secret key communication raises some questions, however: For example:

- How did Computer A and Computer B obtain the shared cryptographic key in the first place?
- If Computer A needs to communicate with many servers and Computer B needs to communicate with many clients, how do they organize the keys that they need for each type of communication?
- How are the keys protected if every Windows 2000 machine needs to store these shared secrets locally?

The answer to these questions lies in the fact that Kerberos authentication is dependent not only on communication between client and server, but also on a third party: the **Key Distribution Center (KDC)**. The KDC is a Windows 2000 service that runs on a physically secure server. It maintains a database of all **security principals**—Windows 2000 computers—in its realm. The KDC maintains a key with each of its security principals to secure its communications. This key is generated from the password of the person or service using the security principal, and by requests from the client-side Kerberos client, immediately after someone logs onto the Windows 2000 computer.

When one security principal needs to communicate with another security principal, the first security principal contacts the KDC and asks for a key for that communication. The KDC generates such a key and sends it to the client computer—Computer A, to return to our earlier example. Computer A's copy of the key is encrypted with the long-term key that it shares with the KDC. Computer B's copy of the key is embedded in a data structure, called a **ticket**, which also includes authentication information for Computer A. The ticket is encrypted with the key shared by the KDC shares with Computer B. It doesn't matter that Computer A can't read the ticket because it doesn't know the key shared by the KDC and Computer B. Instead, Computer A simply stores the ticket in memory and passes it to Computer B when Computer A wants to start the communication. Computer B decrypts the communication with the key that it shares with the KDC and now has a shared key with Computer A.

Certificates

Certificates represent a portable method of authentication for a user or service. Basically, a certificate says, "I am what I say I am." Although certificates don't guarantee anything else about the person or service that they're identifying (for instance, a service with a certificate could potentially have a damaging effect on your computer), they assure you (or, more precisely, Windows 2000) that the identity of the user or service is known. Because certificates are such a trusted source of identification, you must have an equally trusted **certificate authority** to assign them. (A certificate authority is a server entrusted with the task of creating certificates for users and services.)

USER PROFILES AND GROUP POLICIES

When using a modern operating system, people expect to be able to customize the interface. Users want to be able to modify environment settings, and network administrators want to be able to control the appearance and settings available to each user so as to better manage what users can and can't do on their computers. In Windows 2000, **user profiles** and **group policies** are the mechanisms that make this kind of control possible.

User Profiles

You can specify the custom settings defining the user environment and have Windows 2000 save those settings to a file, called a user profile. A user profile in Windows 2000 includes the following information:

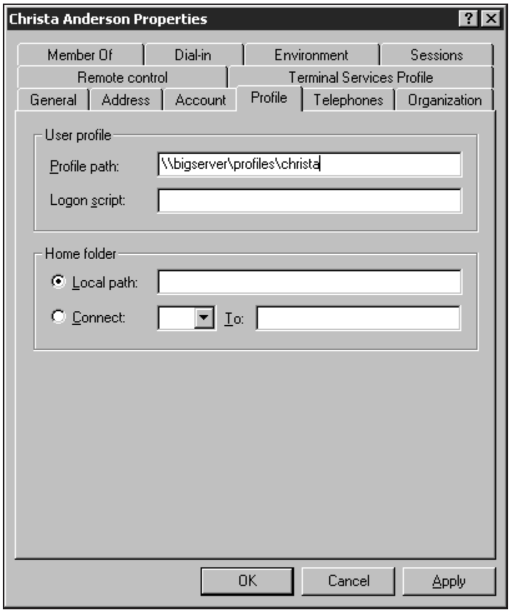
- Screen colors
- Program items
- Network and printer connections
- Mouse settings
- Window size and position
- Redirected folders

Three kinds of user profiles exist: local, roaming, and mandatory. A **local user profile** is created the first time a user logs onto a computer and is stored on the computer's hard disk. The profile is then loaded and defines the user environment whenever that particular user logs onto that computer. If the user changes the profile, the modifications are saved when the user logs off. Users with local user profiles may have different environment settings for each computer they log into.

Roaming user profiles are set up by the system administrator and stored on a network server. When a user logs on, the profile server downloads the profile to the user's computer. As with local profiles, any changes that the user makes to the profile are saved when the user logs off. Users with roaming user profiles always have the same environment settings, no matter where they log into the network.

Mandatory user profiles are roaming user profiles that the user cannot edit. Only the system administrator can edit a mandatory user profile.

A local user profile is the simplest to create—it is created when the user logs onto a computer and then logs off. Roaming and mandatory profiles are a bit more complicated to create, because the system administrator must edit the profile settings for that user account (as shown in Figure 12-3) to point to the directory path from which the profile should be loaded. To assign a roaming profile, you type the network path to the profile's location, followed by the user's logon name, like this: \\servername\profilesfoldername\username. The roaming profile doesn't have to exist at the location you specify, because the user creates it automatically at logon in a folder with the same name as the user's logon name.



Profile path: \\bigserver\profiles\christa

Logon script:

Home folder

● Local path:

○ Connect: To:

Figure 12-3 Configuring the location of a user profile

To assign a mandatory profile, you must create a profile and then point to it. To create a mandatory profile, open the System applet and access the user Profiles tab. Select an existing profile with the desired settings (see Figure 12-4) and copy it to the profiles folder. Once the file exists there, rename it with a .man (for mandatory) extension. Next, return to the Profiles tab of the property sheet for the user account. Type the network path to the profile's location and specify the name of the mandatory user profile, like this: \\servername\profilesfoldername\filename. You don't need to include the file's .man extension unless the domain includes Windows NT 3.x computers.

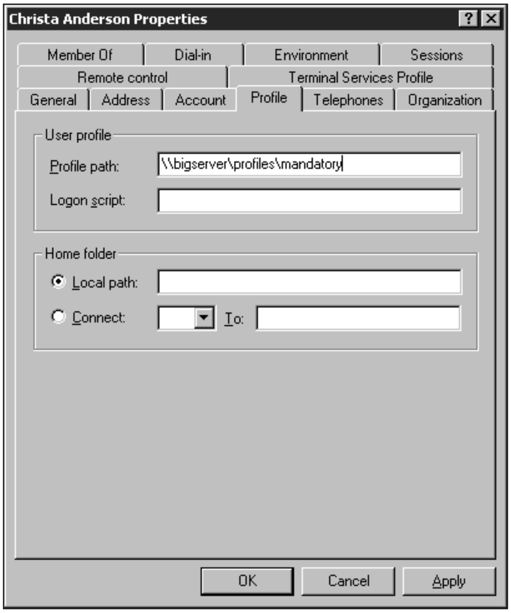
You can also create mandatory profiles for various groups. For example, the mandatory profile for people in the Users group could be Users.man; the mandatory profile for people in the Account Operators group could be Ao.man.

Note

A user can have different user profiles for ordinary network sessions and for terminal sessions. For example, a user might have a roaming network profile, but a mandatory terminal services profile. Configure network client profiles from the Profiles tab of a user's property sheet; configure terminal session profiles from the Terminal Services Profile tab.

Group Policies

Group policies control the security settings for computers and users in the domain. Almost any kind of control you might apply to a user account, user group, or computer in the domain can be set in the group policies.



Profile path: \\bigserver\profiles\mandatory

Logon script:

● Local path:

○ Connect: To:

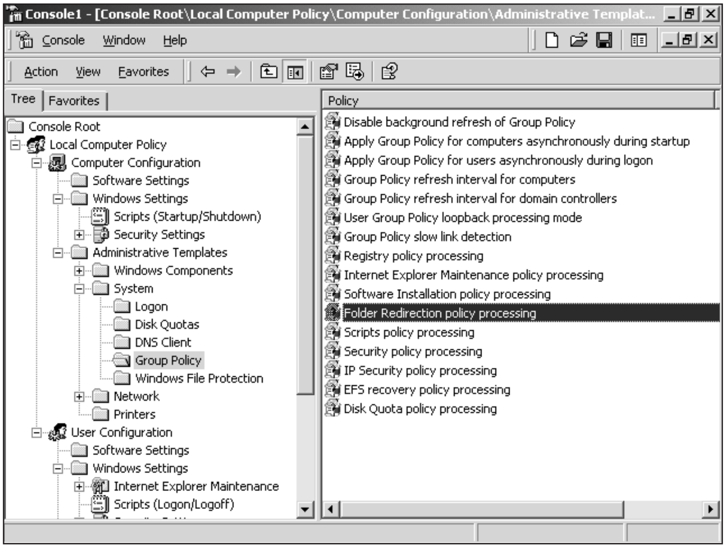
Figure 12-4 Mandatory profiles are based on existing profiles

Types of Group Policies

Group policies for users and computers are classified into three broad categories: software settings, Windows settings, and administrative templates. The exact contents of each policy depend on whether you are talking about a policy applied to a computer or to a user.

- Software settings control how software is installed and performs on the computer or for the user to whom the policy applies.
- Windows settings for users and computers control which logon and logoff scripts are run; they also configure security settings for the policy. For group profiles, the Windows settings control Internet Explorer settings and choice options for the Remote Installation Services, which permit users to install applications over the network. The Windows settings in some group policies might also include a Folder Redirection policy (see Figure 12-5). You can use this policy to point program folders to a new location, either using the same location for all users within that part of the Active Directory or pointing users in different groups to different folders customized to the needs of those group members. For example, you can use this setting to customize the Start menu for each group.

The Administrative Templates section, found in both user and computer group policies, is a comprehensive collection of settings that allow you to manage every part of the user environment that's controlled in the Registry.



Disable background refresh of Group Policy
Apply Group Policy for computers asynchronously during startup
Apply Group Policy for users asynchronously during logon
Group Policy refresh interval for computers
Group Policy refresh interval for domain controllers
User Group Policy loopback processing mode
Group Policy slow link detection
Registry policy processing
Internet Explorer Maintenance policy processing
Software Installation policy processing
Folder Redirection policy processing
Scripts policy processing
Security policy processing
IP Security policy processing
EFS recovery policy processing
Disk Quota policy processing

Figure 12-5 The Folder Redirection policy

The default setting for all group policy options is not configured; in fact, group policies are not applied until you explicitly apply them. You can either enable or disable a group policy, depending on the wording of the policy itself as displayed in the Microsoft Management Console (MMC). For example, to disable the Registry editing tools in a particular user policy, you would enable the policy that said to disable the tools. Disabling the policy not only makes those tools available, but can prevent another policy from making the tools available.

Applying Group Policies

In Windows 2000, group policies are normally applied at some level of the Active Directory structure; from there, they are inherited by the parts of the Active Directory that fall beneath them. For example, the domain might have one group policy, while each of the domain's organizational units (OUs) has another policy. When a user logs onto the domain, the group policy applied to that session depends on the OU of which the computer is part and the OU to which the account belongs. If more than one group policy is associated with a particular OU, site, or domain, the group policy with the highest precedence takes control. Precedence is set from the Group Policy tab of the Properties sheet of the part of Active Directory for which you're establishing a group policy.

Group policies are an extremely complex part of security in Windows 2000. Their characteristics can be summarized as follows:

- Group policies control security and Registry settings for domain use.
- Group policies may be set for either users or computers, or both.
- Group policies are set for each part of a domain. If two policies conflict, the least restrictive policy normally takes control. The only time when this statement doesn't hold true is when policy inheritance is disabled.

DATA ENCRYPTION

One method of protecting data involves encrypting it. **Encryption** is a blanket term for any method of systematically garbling text into a form called **ciphertext** to conceal its meaning. To read it, you must apply an algorithm called a **key** that reverses the logic used to encrypt the text. When you apply the key to the ciphertext, you turn it back to its readable form, called **plaintext**. Only the people who have the key can decrypt the ciphertext. The encryption method may be simple, such as substituting each letter for the one that occurs three letters later in the alphabet, or it may be very complex. The more complex the encryption algorithm, the longer it takes to encrypt and decrypt data, even with the key. Very complicated keys are left to computers, which can quickly perform the calculations required to apply the algorithm.

Two main types of encryption exist: **symmetric encryption** and asymmetric (or **public key**) encryption key. These types of encryption are discussed in the following sections.

Symmetric Encryption

Symmetric encryption uses the same algorithm to encrypt and decrypt text. Kerberos uses symmetric encryption, because clients and servers share a cryptographic key. For files to be used and accessible only to a single person, symmetric encryption is adequate because only a single person needs to know the key. If several people must share a file, however, the situation becomes more complicated. The person doing the encrypting needs to tell the person planning to decrypt the file what the key is, without compromising the key's security. For example, Susan can encrypt data for Fred, but Fred will need the key to decrypt the data and read it. If Susan works in Texas and Fred is located in Wisconsin, getting the key to Fred securely is tricky. Letters, telephone calls, or e-mail can all be intercepted, thereby compromising the key. One solution is to maintain a book that indicates which keys to use under which circumstances. As both sides found out during the course of World War I, such books of this type may be lost.

Another issue with symmetric encryption is the problem of ascertaining who encrypted a file. For example, someone passes an encrypted file to George, indicating that the file came from Susan. The key George has works on the file, so he knows that Susan could have encrypted it. Because Susan shares files with other people, however, Fred could have just as easily created the file while trying to pass it off as Susan's.

Public Key Encryption

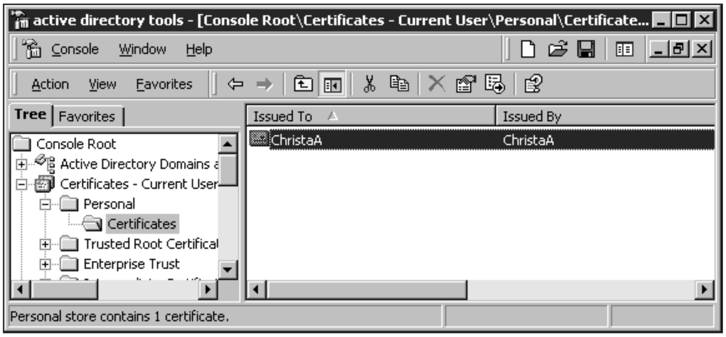
Because of the shortcomings of symmetric encryption, public key encryption has emerged as the de facto standard for encrypting data intended for computer transmission. This encryption method uses two user-specific keys to encrypt and decrypt data: one public and one private. To encrypt data, you must encrypt it with the recipient's public key. The recipient applies that **private key** to decrypt the text. Public keys can be distributed, but private keys are reserved for their owners; there's no such thing as a generic private key. If you want to encrypt data for David, you must use David's public key to do it.

Data Encryption in Windows 2000

Support for encrypted text is not new to Windows. Many encryption tools are available on the Web that you can use with any operating system to secure your data, such as Elgamal, RSA (Rivest, Shamir, and Adelman), Diffie-Hellman, and DSA (Digital Signature Algorithm). Windows 2000 is unique among Windows operating systems, however, in that native encryption is built into its NTFS file system. This feature allows you to secure your documents so that only you—or the people to whom you give the private key—can view the documents. As a result, you can even keep shared documents private or secure files on a computer, such as a laptop, that might easily be stolen. The files are visible to anyone with access to the directories in which they're stored. When someone attempts to open an encrypted file, however, Windows 2000 checks whether the user has a key to that file. If so, the file opens normally. If not, the user is forbidden access to the file. The denial isn't application-dependent. For example, although it's possible to open .doc files in either Microsoft Word or WordPad (the word processor that comes with Windows 2000), you can't open an encrypted .doc file in either application. Without the key to that file, you don't have permission to open it.

Although Windows 2000 supports three file systems (FAT16, FAT32, and NTFS), you can encrypt only files stored on NTFS volumes. This restriction applies because encryption is an NTFS **attribute**, which is a characteristic of a file. FAT file systems under Windows 2000 do not include an encryption attribute, just as they do not include attributes for many of the other advanced features of the NTFS file system, such as file compression and local file security.

When you encrypt data for the first time, you generate a request for a new security certificate that identifies you to Windows 2000. A cryptographic service provider (CSP) generates two 56-bit unrelated keys: a public key, used for encrypting data destined for you, and a private key, used for decrypting that data. The CSP passes the security certificate to the certificate authority, which uses it to create a public key for you. The certificate and public key are stored in the Personal, Certificates folder located in the Certificates add-in to the MMC (see Figure 12-6).


active directory tools - [Console Root\Certificates - Current User\Personal\Certificate...
Console
Window
Help
Action
View
Favorites
Tree
Favorites
Console Root
Active Directory Domains a
Certificates - Current User
Personal
Certificates
Trusted Root Certifical
Enterprise Trust

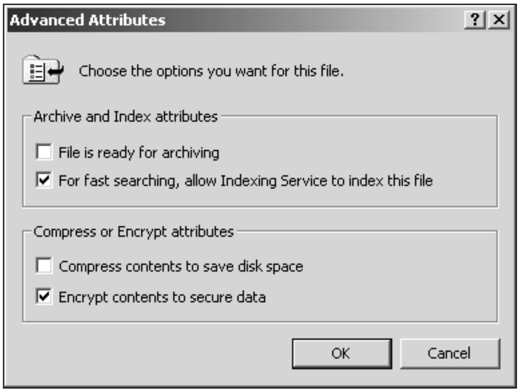
Issued To	Issued By
ChristaA	ChristaA

Personal store contains 1 certificate.

Figure 12-6 Personal encryption certificates are visible from the Certificates add-in

Encryption Tools in Windows 2000

The process of encrypting data in Windows 2000 is very simple. Open Windows Explorer (Start, Programs, Accessories, Windows Explorer). Right-click a file or folder stored in an NTFS directory, and then choose Properties from the context menu. Select the General tab, and click the Advanced button to open the dialog box shown in Figure 12-7. (If you don't see an Advanced button, the file you selected isn't stored in an NTFS volume.)



Advanced Attributes

Choose the options you want for this file.

Archive and Index attributes

☐ File is ready for archiving

☑ For fast searching, allow Indexing Service to index this file

Compress or Encrypt attributes

☐ Compress contents to save disk space

☑ Encrypt contents to secure data

Figure 12-7 Setting encryption attributes for a file

As you can see, two advanced NTFS attributes are available: encryption and compression. The two are mutually exclusive: a file cannot be both encrypted and compressed with NTFS attributes. Check the box that says "Encrypt contents to secure data," and click OK. Click OK again to exit the property sheet. Check the option you want, and click OK. If the folder containing the file is not encrypted, you will be warned of this fact before you close the property sheet and prompted to encrypt both the folder and the file. Because new files inherit the encryption attributes of the folder in which they're stored, it is a good idea to store only encrypted files in encrypted folders. Otherwise, if the file changes in a way that makes Windows 2000 perceive it as new, the file will lose its encryption attribute.

Moving and copying files between encrypted and plaintext directories introduces some problems. If you copy or move an unencrypted file to an encrypted NTFS folder, the file becomes encrypted. If you copy or move an encrypted file to an unencrypted NTFS folder, the file remains encrypted. If you copy or move an encrypted file to a FAT or FAT32 folder, the file is no longer encrypted (because encryption is an NTFS attribute). As a security measure, Windows 2000 does not allow you to copy an encrypted file to an unencrypted folder unless you have the private key needed to decrypt the file.

Windows 2000 users can encrypt data across the network, but the certificate always remains on the Windows 2000 machine where the ciphertext is stored. The private key is stored in the Registry of the computer holding the ciphertext.

Windows 2000 also supports a command-line encryption utility called CIPHER. To encrypt a single folder in the current directory, type **cipher /e *foldername***, where "foldername" is the name of the folder you want to encrypt and "/e" stands for "encrypt." To decrypt the same folder, replace the /e switch with /d, like this: **cipher /d *foldername***. The command will report whether the operation succeeded. Because compressed data can't be encrypted, if you try to encrypt a compressed file or folder from the command line, you will receive an Access Denied error. You must uncompress the file or folder before Windows 2000 can apply the encryption attribute to the file object.

Tip

Windows 2000 can't encrypt or decrypt a folder's contents if they're in use (or even displayed in a Windows Explorer window), so make sure that no one is using files that you are attempting to encrypt or decrypt.

Windows 2000 makes the process of decrypting a file transparent to the person who uses the file. Whenever you attempt to open the file, the encryption file system checks whether a private key belonging to you is stored on the computer. If it is, you can open the file. If it is not, you are denied access.

Removing the encryption attribute altogether is another matter. To decrypt a file or folder to allow anyone to read it, right-click the file, choose Properties, select the General tab, and click the Advanced button. In the Advanced Attributes dialog box, uncheck the "Encrypt contents to secure data" check box (refer to Figure 12-7). The file will then be open to anyone with the permission to access it.

Enforcing Encryption

Encrypted files don't have any obvious differences alerting people to their off-limits status. Indeed, the contents of an encrypted folder are displayed like any other shared or locally available data. The only way you can tell that you have attempted to open an encrypted file is that you won't be able to open it. If you try, the server will chug away for a moment or two while Windows 2000 looks for a private key that matches your certificate, then the system will deny access when it doesn't find one. Not even administrators can open or decrypt files that someone else has encrypted. In addition, file ownership is not relevant to decryption—if a person takes ownership of someone else's encrypted file, the new owner will not be able to open it.

Protecting Encryption Keys

Microsoft implemented the Windows 2000 encryption services especially for the benefit of laptop users who wanted to keep their data secure even if their laptops were stolen. This security plan has a major hole, however. If someone steals a laptop and you haven't protected your account, the thief can read your encrypted files. To avoid this problem, Microsoft recommends exporting each user's certificate with the private key and saving it to disk, then deleting the certificate on the computer. That way, even if intruders log onto the computer with your account, they won't be able to read the encrypted files stored there.

Chapter Summary

- To secure your network, you need to know how you can use the operating system to control access to network resources. Two main modes of licensing exist: per-seat and per-user. Different types of licenses are also available: console license, client access license, terminal server license, and application license.
- After your licenses are in place, you need to assign user and group permissions, which you do via server or domain accounts and various methods of authentication, such as NTLM and Kerberos. You can also control security by managing your user profiles and group policies and by using certificates and data encryption. Two types of encryption exist: symmetric and public key. You can manage encryption in Windows 2000 via Windows Explorer or through the CIPHER command-line utility.

Key Terms

application license — A license that allows you to run a particular application.

application service provider (ASP) — A service running applications from a terminal server and making them available to anonymous users via the Internet for a fee.

attribute — A characteristic associated with a file object (file or folder). Different file systems have different attributes.

authentication — The process that a computer undertakes to determine that you are who you say you are.

certificate — A portable method of authentication that demonstrates the identity of a user or service. Certificates are files that may be imported or exported, so you can move or copy them if necessary.

certificate authority — A server entrusted with the task of creating certificates for users and services.

ciphertext — Encrypted data.

client access license — A type of license that permits the holder to access a server from the network.

console license — A type of license that comes with an operating system and represents permission to install the operating system on a single machine and use it from that machine.

domain — A group of computers that shares a centralized security database.

domain controller — The computer that stores the domain's security database. A domain can have more than one domain controller to ease the burden of authenticating users.

encryption — A blanket term for any method of systematically obscuring the meaning of data by applying an encryption key to it.

End User License Agreement (EULA) — Paper or software text accompanying software that defines the conditions under which the licensee may use the software.

group policies — Policies that control the security settings for computers and users in the domain.

groupware — Multiuser applications that come with a group license and are used by several people simultaneously, such as e-mail servers.

interactive logon — The act of typing your name and password into the login screen of a Windows 2000 computer.

Internet client license (ICL) — A type of license that permits an anonymous user to log onto a terminal server via the Internet. ICLs are restricted for anonymous use; people with domain accounts can't use them.

Kerberos — The native Windows 2000 authentication protocol. Kerberos relies on a system of shared secrets for mutual authentication of client and server.

key — An algorithm used to encrypt or decrypt data. Sometimes, the same key may do both; at other times, the encryption key may be different from the decryption key.

Key Distribution Center (KDC) — A secure server in a Windows 2000 domain that's responsible for generating the cryptographic keys and tickets that are the basis of Kerberos security.

Local Security Authority (LSA) — The component that checks whether a user logging on has an account on a local or a trusted domain. When you are logging onto another domain, the LSA must communicate with that domain's domain controller to see whether the domain controller has an account for you in its security database.

local user profile — A user profile stored on the local computer; the default setting for all user profiles. Local user profiles exist on a per-computer basis, so a user may have different environment settings depending on which computer he or she logs onto. Changes to the profile are saved to the local computer when the user logs off.

mandatory user profile — A roaming user profile that is not user-definable. If the user changes the environment settings, those changes are not saved at logoff. A mandatory user profile has a .man extension.

NT LAN Manager (NTLM) — The default authentication protocol used in Windows NT 4.

per-seat license — A type of license that permits a predefined number of computer connections to the operating system or application being licensed.

per-session license — A type of license that permits a predefined number of simultaneous user connections to the operating system or application being licensed.

plaintext — Unencrypted data.

private key — A key devoted to decrypting data for a particular person. Private keys should be kept secure.

public key — A key devoted to encrypting data for a particular person. A public key only encrypts; it does not decrypt.

roaming user profile — A user profile stored on a network server and downloaded to whichever computer a user is currently logged into. Changes to the profile are saved to the network server when the user logs off.

secret key communication — The method of authentication on which Kerberos is based, where a client and server must both know and use the same cryptographic key to protect the network.

security principal — A Windows 2000 computer in a domain using Kerberos.

symmetric encryption — A method of data encryption that uses the same algorithm to encrypt and decrypt plaintext.

terminal server client access license (TSCAL) — A type of license that permits the computer to which it's assigned to run a session from a terminal server.

ticket — A data structure generated by the KDC when a client computer asks the KDC for a secret key. The server's half of the secret key is embedded in the ticket and encrypted with the key that the KDC and the server have in common.

user profile — A file containing environment settings, which is loaded when a person logs onto a computer or domain. User profiles may be stored on the local computer or on a server, and may be either user-definable or locked down.

Review Questions

1. What is the Windows 2000 file system that supports native encryption?
   a. FAT32
   b. NTFS
   c. FAT
   d. Kernel
2. ________________ licenses permit a certain number of simultaneous connections to the operating system or application being licensed.
3. Per-seat licenses are assigned to users. True or False?
4. You're running one instance of WordCruncher on your local computer (running Windows 98) and one in a terminal session on the same computer. WordCruncher is licensed on a per-seat basis. You need one application license. True or False?
5. Does your answer to question 4 change if your computer is running Windows 2000 Professional? Why or why not?
6. Which of the following describes the interactive logon process?
   a. When you type your name and password, the Local Security Authority passes the information to WinLogon, which then communicates with the domain controller to see whether you have a valid account.
   b. When you type your name and password, the domain controller generates a certificate that authenticates you on the domain.
   c. When you type your name and password, the Local Security Authority generates a secure key from the password and passes that key to the domain controller to secure communications.
   d. When you type your name and password, WinLogon accepts the information and passes it to the Local Security Authority, which compares the information with the contents of the domain security database.

7. Which authentication protocols for LANs does Windows 2000 support? (Choose all that apply.)

   a. TCP/IP

   b. Kerberos

   c. IPX/SPX

   d. NTLM

8. The native Windows 2000 authentication protocol works with a system of shared secrets. Both the client and the server know the same cryptographic key, and must prove their identities by sending the key across the network before any communication can begin. True or False?

9. Which of the following is part of the native Windows 2000 authentication process? (Choose all that apply.)

   a. Clients

   b. Protocols

   c. KDC

   d. Server

10. ________________ can define the contents of your Documents folder; ________________ can define whether you have a Documents folder and redirect it to a new location.

11. By default, what are the per-user environment settings?

    a. Stored on the local computer

    b. Stored on the domain controller

    c. Stored in the Active Directory

    d. None of the above

12. Kerberos uses symmetric encryption. True or False?

13. If you take ownership of someone else's encrypted file, can you read the file? Why or why not?

14. To access a Windows 2000 terminal server from Windows 2000 Professional, what do you need?

    a. A CAL

    b. A TSCAL

    c. A groupware license

    d. All of the above

15. Which of the following is not a reason to have multiple domain controllers in the domain? (Choose all that apply.)

    a. Having multiple domain controllers reduces network traffic.

    b. Having multiple domain controllers may make user authentications faster.

c. Having multiple domain controllers reduces the amount of time required for Kerberos authentication.

d. Having multiple domain controllers reduces the stress on the main domain controller.

16. Which of the following statements about the Local Security Authority are true?

a. You don't need it in a Windows 2000-only network because Kerberos fulfills the same function.

b. It authenticates you both at the computer you log onto and when you access a network resource in a mixed network.

c. Both a and b

d. Neither a nor b

17. Which of the following operating systems support Kerberos?

a. Windows NT and Windows 2000

b. Windows NT Server Enterprise Edition and Windows 2000

c. Windows 2000 Server

d. Windows 2000

18. Which of the following describes Kerberos?

a. Kerberos requires a central server.

b. Kerberos does not require you to be reauthenticated on the network each time you access a new resource.

c. Kerberos uses private key encryption to prove client and server identities.

d. To make Kerberos work, you need a central server to pass out tickets.

19. Which of the following cannot be set with a mandatory user profile?

a. Window size and position

b. Program items

c. Network and printer connections

d. None of the above

20. Group policies may be applied on many different levels of the Active Directory structure. If policy settings for a particular person conflict, which policy takes precedence?

a. The domain level

b. The user level

c. The group level

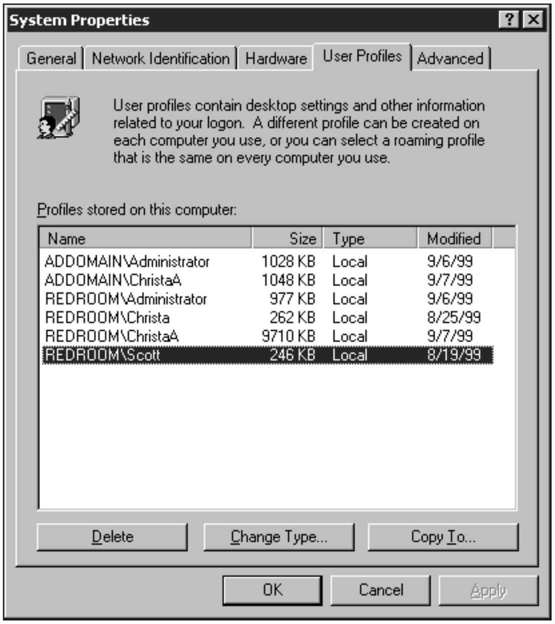
d. The least restrictive policy controls

HANDS-ON PROJECTS

Project 12-1

To copy a profile to a new location and make it a mandatory profile:

1. Open the **System applet** in the **Control Panel** and select the **User Profiles** tab, as shown in Figure 12-8. Select a profile not currently in use.



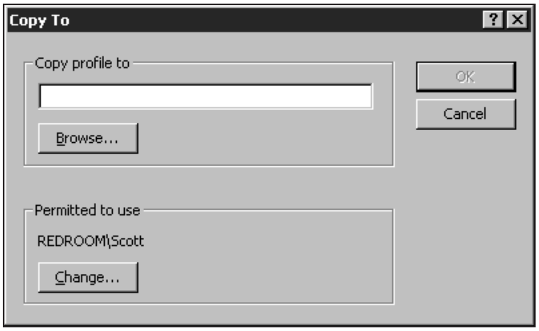
User profiles contain desktop settings and other information related to your logon. A different profile can be created on each computer you use, or you can select a roaming profile that is the same on every computer you use.

Profiles stored on this computer:

Name	Size	Type	Modified
ADDOMAIN\Administrator	1028 KB	Local	9/6/99
ADDOMAIN\ChristaA	1048 KB	Local	9/7/99
REDROOM\Administrator	977 KB	Local	9/6/99
REDROOM\Christa	262 KB	Local	8/25/99
REDROOM\ChristaA	9710 KB	Local	9/7/99
REDROOM\Scott	246 KB	Local	8/19/99

Figure 12-8 The User Profiles tab displays all user profiles stored on the local computer

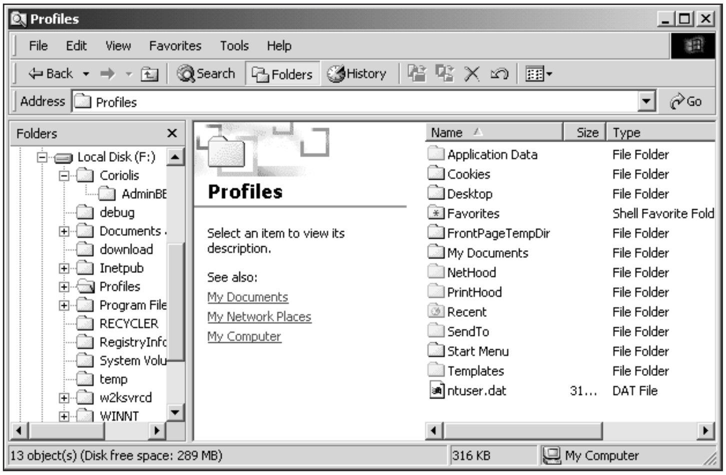
2. Click the **Copy To** button to open the dialog box shown in Figure 12-9.



REDROOM\Scott

Figure 12-9 Choose a shared folder in which to store the profile

3. Click the **Browse** button and choose a shared folder in which to put the profile. To keep things simple, it is best to name this folder "Profiles."
4. Click **OK**, then click **Yes** to confirm the copy operation. Finally, click **OK** to exit the system applet.
5. In Windows Explorer, browse to the folder where you copied the file. Set the view so that you can see hidden files. Look for a file named Ntuser.dat (see Figure 12-10) and rename it with a .man extension.



Profiles

Select an item to view its description.

See also:
My Documents
My Network Places
My Computer

Name	Size	Type
Application Data		File Folder
Cookies		File Folder
Desktop		File Folder
Favorites		Shell Favorite Fold
FrontPageTempDir		File Folder
My Documents		File Folder
NetHood		File Folder
PrintHood		File Folder
Recent		File Folder
SendTo		File Folder
Start Menu		File Folder
Templates		File Folder
ntuser.dat	31...	DAT File

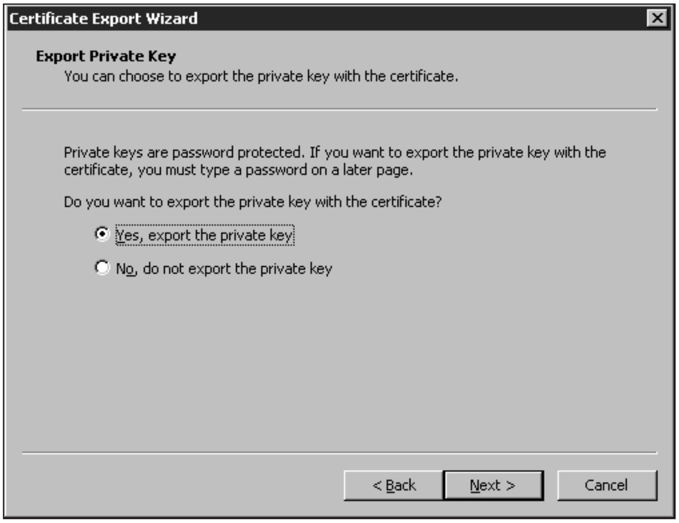
13 object(s) (Disk free space: 289 MB) 316 KB My Computer

Figure 12-10 Rename a profile file with a .man extension to make it mandatory

Project 12-2

To export a certificate and private key from one Windows 2000 computer and import it to another computer:

1. In the MMC (**Start**, **Run**, **MMC**), add the **Certificates snap-in** (be sure to select the **My User Account** option).
2. In the **Personal** folder, open the **Certificates** folder. The per-user certificates on the computer are displayed in the right pane. Right-click the certificate that you want to export as a file, and choose **Export** from the **All Tasks** menu. The Certificate Export Wizard opens.
3. Click past the opening screen to the first screen (Figure 12-11), which asks whether you would like to export the private key along with the certificate. You need the private key to decrypt data, so select it.



Export Private Key

You can choose to export the private key with the certificate.

Private keys are password protected. If you want to export the private key with the certificate, you must type a password on a later page.

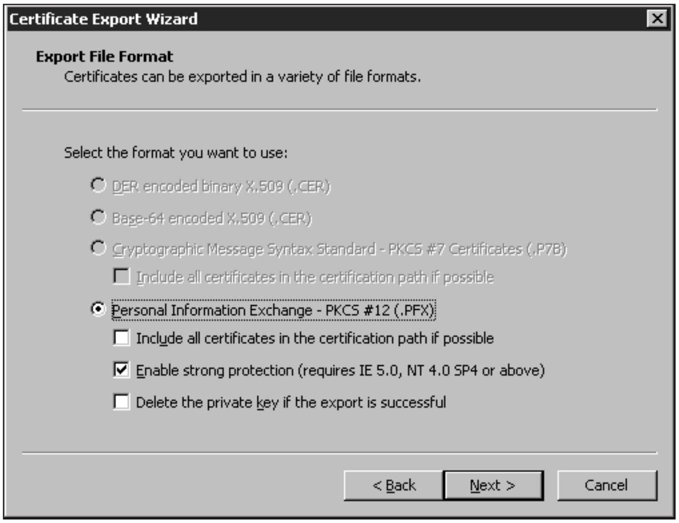
Do you want to export the private key with the certificate?

● Yes, export the private key

○ No, do not export the private key

Figure 12-11 Export the private key with the certificate

4. In the next screen, shown in Figure 12-12, choose the export options, including the file type, the strength of encryption, and action to take with the local key if the export works.



Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

○ DER encoded binary X.509 (.CER)

○ Base-64 encoded X.509 (.CER)

○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)

□ Include all certificates in the certification path if possible

● Personal Information Exchange - PKCS #12 (.PFX)

□ Include all certificates in the certification path if possible

☑ Enable strong protection (requires IE 5.0, NT 4.0 SP4 or above)

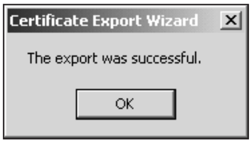
□ Delete the private key if the export is successful

Figure 12-12 Set certificate export options

5. In the next screen of the wizard, supply a password to import the certificate and private key. Choose this password carefully.
6. Choose a filename for the key, by either typing a path or browsing for it. Although you can save the file on any volume, NTFS volumes are generally a more secure storage

location if you don't need the flexibility of a floppy disk. Save the certificate on a floppy disk or, better yet, in a safe network location where it can be backed up, then delete it from the computer. You'll be able to open encrypted files, but the certificate will no longer reside on the machine.

7. The final screen of the wizard displays your choices. Review them carefully, then click **Finish** to export the keys. If the export operation worked, Windows 2000 displays a message box to tell you so (see Figure 12-13).


Certificate Export Wizard

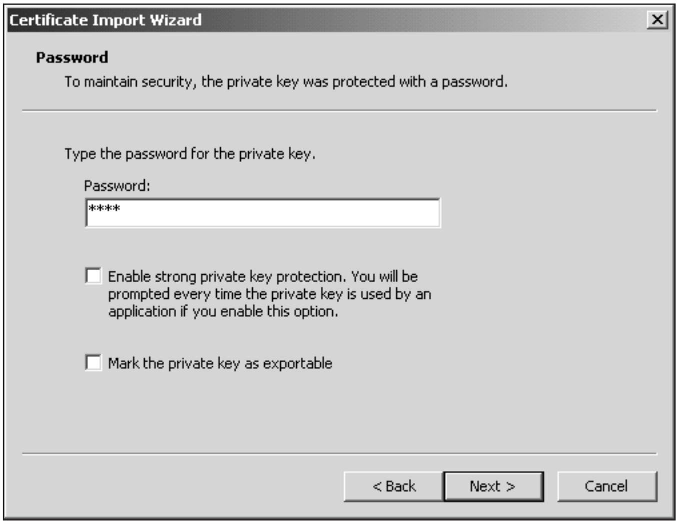
The export was successful.

Figure 12-13 Successful certificate export confirmation

Project 12-3

To import a certificate to a new computer:

1. To import the certificate to another computer or replace it on the same one, open the **Personal** folder, right-click the **Certificates** folder, and choose **Import** from the **All Tasks** list. The Certificate Import Wizard opens.
2. Click **Next** on the opening screen, then browse for the file you saved in Hands-on Project 12-2.
3. If the certificate you're importing includes the private key, you need to supply the password assigned when the key was exported. Type it as shown in Figure 12-14, and choose the degree of control you want over the private key.


Certificate Import Wizard

Password

To maintain security, the private key was protected with a password.

Type the password for the private key.

Password: ****

□ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.

□ Mark the private key as exportable

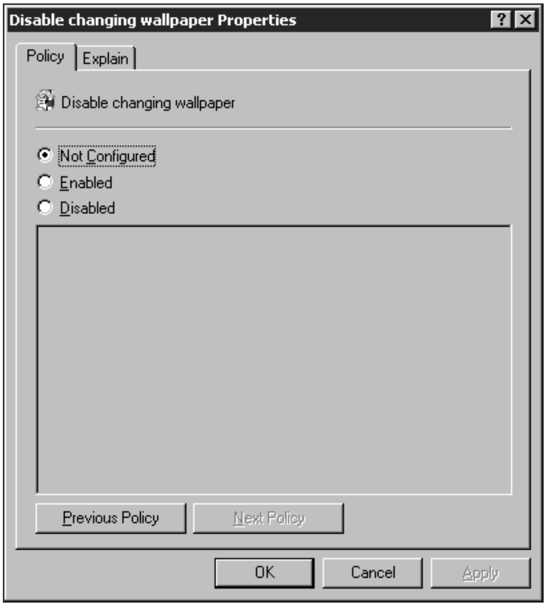
Figure 12-14 Provide the password to the private key and choose the degree of control

4. Specify where the new key should go. For user keys, the Personal folder should be fine.
5. Review the importing options, then click the **Finish** button to import the key. Windows 2000 will tell you whether the importing action succeeded.

Project 12-4

To edit a group policy setting:

1. In the MMC (**Start**, **Run**, **MMC**), add the **Group Policy snap-in** for the local computer. You should see an icon for the Group Policy snap-in.
2. Click **Local Computer Policy** in the left pane of the MMC to reveal the User Configuration and Computer Configuration objects within it. Double-click the **User Configuration** object to show its contents.
3. Open the **Administrative Templates** folder, the **Control Panel** folder, and then the **Display** folder.
4. Double-click **Disable changing wallpaper** to open the property sheet shown in Figure 12-15. Click the **Explain** tab to see an explanation of this policy.



Disable changing wallpaper

● Not Configured
○ Enabled
○ Disabled

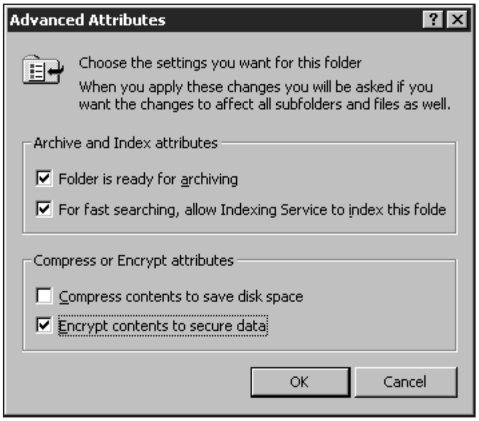
Figure 12-15 View the properties for a group policy

5. From the **Policy** tab, click **Enable**, and then click **OK**. Users to whom this policy applies will not be able to change their system wallpaper.

Project 12-5

To encrypt a folder from Windows Explorer:

1. Right-click a folder stored in an NTFS directory, and then choose **Properties** from the context menu to open the folder's property sheet.
2. On the **General** tab, click the **Advanced** button to open the dialog box shown in Figure 12-16. (If you don't see an Advanced button, the file you selected isn't stored in an NTFS volume.)



Choose the settings you want for this folder

When you apply these changes you will be asked if you want the changes to affect all subfolders and files as well.

Archive and Index attributes

☑ Folder is ready for archiving

☑ For fast searching, allow Indexing Service to index this folde

Compress or Encrypt attributes

☐ Compress contents to save disk space

☑ Encrypt contents to secure data

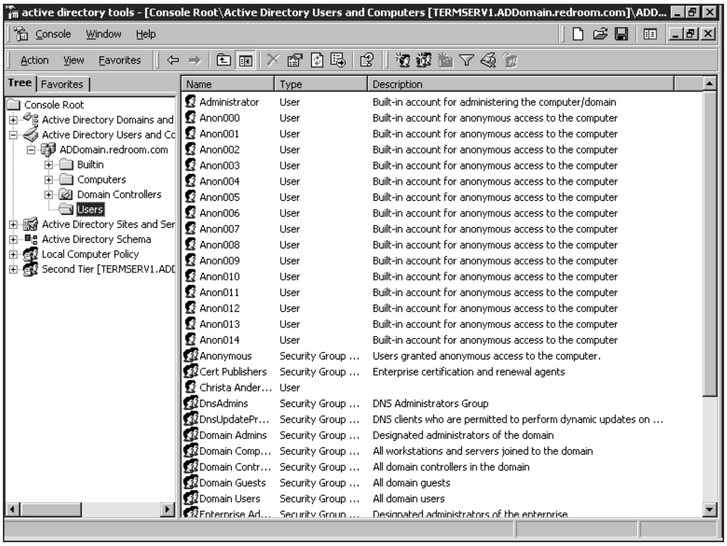
Figure 12-16 Encrypting a folder

3. Two advanced NTFS attributes appear here: encryption and compression. The two are mutually exclusive; a file cannot be both encrypted and compressed with NTFS attributes. Click the box next to **Encrypt contents to secure data**, and then click **OK**.
4. Click **OK** again to exit the property sheet. Click **OK** again to confirm the attribute change. The file is now encrypted.

Project 12-6

To specify a new path for a roaming user profile:

1. From the **Active Directory Users and Computers** snap-in to the MMC, open the **Users** folder as shown in Figure 12-17.

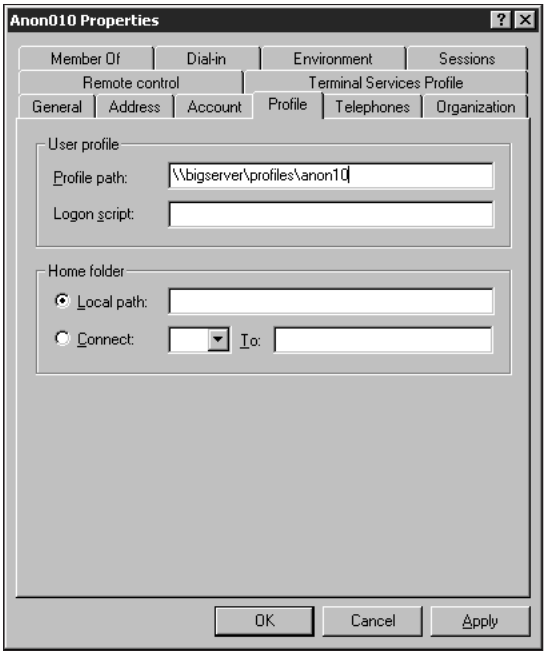


active directory tools - [Console Root\Active Directory Users and Computers [TERMSERV1.ADDomain.redroom.com]\ADD...

Name	Type	Description
Administrator	User	Built-in account for administering the computer/domain
Anon000	User	Built-in account for anonymous access to the computer
Anon001	User	Built-in account for anonymous access to the computer
Anon002	User	Built-in account for anonymous access to the computer
Anon003	User	Built-in account for anonymous access to the computer
Anon004	User	Built-in account for anonymous access to the computer
Anon005	User	Built-in account for anonymous access to the computer
Anon006	User	Built-in account for anonymous access to the computer
Anon007	User	Built-in account for anonymous access to the computer
Anon008	User	Built-in account for anonymous access to the computer
Anon009	User	Built-in account for anonymous access to the computer
Anon010	User	Built-in account for anonymous access to the computer
Anon011	User	Built-in account for anonymous access to the computer
Anon012	User	Built-in account for anonymous access to the computer
Anon013	User	Built-in account for anonymous access to the computer
Anon014	User	Built-in account for anonymous access to the computer
Anonymous	Security Group ...	Users granted anonymous access to the computer.
Cert Publishers	Security Group ...	Enterprise certification and renewal agents
Christa Ander...	User	
DnsAdmins	Security Group ...	DNS Administrators Group
DnsUpdatePr...	Security Group ...	DNS clients who are permitted to perform dynamic updates on ...
Domain Admins	Security Group ...	Designated administrators of the domain
Domain Comp...	Security Group ...	All workstations and servers joined to the domain
Domain Contr...	Security Group ...	All domain controllers in the domain
Domain Guests	Security Group ...	All domain guests
Domain Users	Security Group ...	All domain users
Enterprise Ad...	Security Group ...	Designated administrators of the enterprise

Figure 12-17 Edit user settings from the Management Console

2. Right-click one of the anonymous accounts and choose **Properties** from its context menu. Select the **Profile** tab, as shown in Figure 12-18.



Anon010 Properties

User profile

Profile path: \\bigserver\profiles\anon10

Logon script:

Home folder

● Local path:

○ Connect: To:

Figure 12-18 Manipulating user account properties

3. Type the path to the shared profiles directory, ending the path with the name of the account. For example, for Anon10, the path might look like this: \\bigserver\profiles\anon10. Do not refer to the profile location with a drive letter name, because that technique works only if the shared folder is always mapped to the same drive letter on all computers.
4. Click **OK** to save the change. Anon10 now stores his user profile in the location you specified.

CASE PROJECTS

1. You have set up a domain using all Windows 2000 Servers for domain controllers. All network servers are running Windows 2000. The clients are running Windows NT Workstation. Which authentication protocol will you use, and why?
2. You're logging onto a Windows 2000 terminal server from a Windows 98 computer. How many licenses must you purchase to make this setup legal? How does the answer change if you're running the terminal session from a Windows 2000 computer?
3. Describe briefly how Windows 2000 applies group policies if a user logs into an OU with one group policy but that OU is part of a domain with another group policy, and the policies conflict in some way. Which policy will maintain control?
4. Describe how you would encrypt data for someone—and how that person would decrypt it—in a public key encryption system.